
ClickDOS II v2.00 - by Gary Scott Yates

[Aminet.net has a Version 1.10 of ClickDOS, that is both limited in its function and differs from the Version 2+ described in these pages. Both versions were released for KS/WB 1.3 and due to the lack of a "NULL device, several commands will not work with 2.04+]

[N.B., Use the Right Mouse Button to enter directories, unlike DOpus's double-click method]

Copyright (C) 1989,1990 Gary S. Yates -- 1200 Happy Hollow Rd. Apt 507, West Lafayette, IN. 47906

New features are marked with **V2.00** in the left margin.

WHAT IS ClickDOS II:

ClickDOS II is an AmigaDOS enhancement utility. It allows you to do DOS-like actions with the mouse instead of typing commands at a Command Line Interface (CLI). It adds a great deal of functionality for the "Click-n-Go" user which is not available in the Workbench environment. It also makes some of the tasks which are very cumbersome to do at a CLI or Workbench, such as moving or deleting a large number of files, much easier to perform.

ClickDOS II is a dream for "cleaning up" floppy or hard drives, or for browsing through disks of public domain software sorting out what you wish to keep. These are, however, only a few of the many uses for ClickDOS II.

INSTALLING ClickDOS II:

ClickDOS II is easy to install. There are five files which ClickDOS II needs to operate correctly:

- The file 'ClickDOS_II' can be located just about anywhere. However, it is best suited for the 'Utilities' directory on your Workbench disk. You can then start ClickDOS II from Workbench with icons or you can start it from your Startup-Sequence. If you wish to be able to start it from a CLI, however, you will need to add the Utilities directory to your path. See your AmigaDOS manual for information on the PATH command.

V2.00 - The files Spool and Spawn need to be in the directory which contains your DOS commands, usually the C directory of your Workbench disk where C: is assigned. (these files only work on KickStart 1.3 and below).

The file ClickDOS-StartUp must be in the directory to which you have S: assigned. Usually S: is assigned to the S directory on your Workbench disk.

V2.00 - The arp.library file must be in the directory to which LIBS: is assigned. LIBS: is usually assigned to the libs directory on your Workbench disk.

In addition to these files, in order for EDIT, RUN and User Defined gadgets to operate, you will need to use the NULL: device which is provided on the ClickDOS II distribution disk. **The NULL: device file only works on KickStart 1.3 and below**). To use the NULL: device:

- The file 'null-handler' needs to be in the L directory of your Workbench disk.
- An entry needs to be added to your MountList which looks like

```
this:
NULL: Handler = L:null-handler
      Stacksize= 500
      Priority = 5
      GlobVec = 1
```

#

- The following line needs to be placed in your startup-sequence:

mount NULL:

You can copy these files by hand from the ClickDOS II distribution disk and do the setup yourself or you can use the INSTALL script provided. The INSTALL script will install ClickDOS II and the NULL: device as described above. To use the INSTALL script:

If you are using AmigaDOS V1.3 you can run Install from the workbench (Iconx must be in your path):

1. Make sure the distribution disk is in a disk drive.
2. Double-click on the ClickDOS II disk icon.
3. Double-click on the ClickDOS_II_INSTALL icon.

If you are using any version of AmigaDOS other than V1.3 (i.e., 1.1 or 1.2), the install script must be run from the CLI DF0:>execute install

The INSTALL script will install ClickDOS II for you. It will copy the files to your Workbench disk as described above and append lines to your MountList and to your startup-sequence. If you do not have a MountList or startup-sequence INSTALL will create one.

Regardless of which method of executing INSTALL you use, when INSTALL is complete, you will probably need to edit your startup-sequence and move the lines which INSTALL added to a better position than at the end of the file. You do NOT have to edit your MountList unless you keep your MountList entries in a specific order for some reason. You do NOT have to re-boot after running INSTALL. It will start the NULL: device for you so that you can begin using ClickDOS II immediately.

USING ClickDOS II:

ClickDOS II can be started from either WorkBench or a CLI.

From a CLI type: **ClickDOS_II [Dir1] [Dir2] [-i]**

Dir1 and dir2 are directories you want ClickDOS II to put into its directory areas as soon as it starts. These are optional and are simply a convenience. **THE -i OPTION** will be explained in the section on iconification. If you do not want ClickDOS II to take over your CLI use the ARP ARUN command.

From WorkBench simply double click on the ClickDOS_II icon.

THE ClickDOS II WINDOW:

Included on the ClickDOS II distribution disk is an IFF picture file called ClickDOS.pic. This picture shows the ClickDOS II window, with labels pointing out the different areas and gadgets. You may find it quite helpful to view this picture to become familiar with the parts of the ClickDOS II window before continuing with this section. Users of ClickDOS V2.06 will find that the ClickDOS II window is significantly different. You may also find it helpful if you start ClickDOS II now and experiment with each feature as it is described here.

The main features of the ClickDOS II window are the two directory areas. These areas are where ClickDOS II shows you the files of the currently selected directories and where you pick the files you want to affect with the action gadgets. You can scroll the file names in a directory area using

the proportional gadget to the right of each area. At the top of the window is a space where ClickDOS II will present messages to you. **WATCH THIS AREA!**

In the upper right-hand corner of the window ClickDOS II will continuously show you how much memory is available in your system. It displays CHIP and FAST memory separately in Kilobytes (K). This display is updated about every second.

DIRECTORY AREA DISPLAY MODES:

The directory areas of ClickDOS II can show you more than just the names of files. They can also show you file sizes, creation dates, protection bits, etc. The information the directory areas display depends on the display mode the directory area is in. The display modes are:

S - display file size in bytes

D - display file creation date

T - display file creation time

P - display protection bits

C - show if a file has a comment

N - show nothing but the file names

B - display file size in blocks

You can change the display mode of a directory area by clicking on the display mode gadgets in the center of the ClickDOS II window. There is a set of display mode gadgets for each directory area. The letter of the current display mode is highlighted for each area.

READING DIRECTORIES:

There are several ways you can get ClickDOS II to read a directory into one of its directory areas and still more ways to manipulate the directory once it has been read.

The quickest way to read a directory is to simply click on one of the device gadgets above each directory area. This will cause ClickDOS II to read the root directory of that device.

There is a set of seven device gadgets for each directory area. The devices represented in these gadgets can be set in the ClickDOS-Startup file explained later.

V2.00 : Another way to get ClickDOS II to read the root directory of a device is to click the Volume Requester Gadget. Clicking this gadget brings up the Volume Requester. This requester will contain every device, volume and assignment (made with the Assign command) in your system. You can select any of these items and ClickDOS II will read the root of that volume. You can also select the Cancel option if you change your mind.

Another way to get ClickDOS II to read a directory is by typing the FULL PATH of a directory name into the Current Directory Gadget located over each directory area. This is a normal text gadget. ClickDOS II will always show the name of the current directory in this gadget.

You can force ClickDOS II to re-read a directory from disk by clicking on the Re-Read gadget. You may want to do this, for example, if you do something at a CLI which changes the contents of the directory. ClickDOS II cannot detect such a thing and you would have to force it to re-read the directory to see the changes.

When ClickDOS II reads a directory, it sorts the contents of the directory alphabetically, files separate from directories. It places all directory names at the top of the list followed by file names. Directory names are shown in the Workbench Screen's highlight color (orange in the default Workbench colors) and file names in the text color (white in the default Workbench colors). The colors ClickDOS II uses on your system will depend on your current preferences settings.

File Comments are a very nice feature of AmigaDOS. Unfortunately because they are so clumsy to set or view with the CLI or Workbench, few people use them. ClickDOS II makes file comments much easier to use. To make you aware that file comments are present, when ClickDOS II reads a directory, if any of the files have comments, ClickDOS II will put the following in the message display:
File(s) read have comments.

MANIPULATING DIRECTORIES:

Once ClickDOS II has read the contents of a directory, you can read subdirectories of that directory by clicking on the name of a subdirectory and either clicking on the Child gadget or clicking the RIGHT mouse button (the menu button) over the directory name. Note, this has been changed since version 2.06 which allowed you to click the right button anywhere over the directory area once you had selected the name. You must now click ON the directory name with the right button. The Double-Click was not used to read subdirectories because of the conflict with clicking on a directory name a second time to deselect it.

You can very quickly read a subdirectory of one directory area into the OPPOSITE directory area by clicking and holding on the directory name then dragging the mouse to the other directory area and releasing the button.

You can move from a subdirectory back to its parent by clicking the Parent gadget. If you click the parent gadget when at the root of a device, ClickDOS II will completely clear the directory area and free all memory it was using to store the filenames. You can cause ClickDOS II to free all memory used by a directory area very quickly by clicking the RIGHT mouse button over the Parent Gadget. You may want to do this for both directory areas to free up as much memory as possible if you are going to put ClickDOS II in the background (iconification is explained below) and use another program.

There are three gadgets located between the directory areas called the Swap Gadgets. These gadgets allow you to do swapping of the contents of the directory areas. The single arrow Swap Gadgets [<] and [>] cause ClickDOS II to read the same directory into the opposite directory area. For example if you have read a directory into the Left area you can quickly have ClickDOS read the same directory into the right area by clicking the Right Swap Gadget [>].

The double Swap Gadget [<>] causes the two directory areas to switch contents. What was in the right area will be read into the Left and vice-versa.

Note that using the swap gadgets deselects selected items.

V2.00 ClickDOS AND WILD CARDS:

ClickDOS II supports all AmigaDOS wild cards and the ARP asterisk (*) wild card. See your AmigaDOS manual and the ARP documentation for a detailed description of these wild cards. You use wild cards by typing them into the Current Directory Gadget. When you enter a wild card into this gadget the directory is read but ClickDOS II only displays the items which match the wild card. For example, if you enter:

DF0:*.info

ClickDOS II will only display .info files in the directory area. Once you have entered a wild card in a Current Directory

Gadget the wild card will remain active even if you move around to other directories. This allows you to scan a set of directories and only see the items which match the wild card. Note, that wild cards affect directories as well as files. You will only see directories that match the wild card.

ClickDOS II AND DISK SPACE:

Above each directory area is a Bytes Selected/Free Space display. In this display ClickDOS II will always show the free space available on the disk associated with the directory in the directory area. The disk free space will be displayed as F:XXXXXXXXXX where XXXXXXXXXXXX is the free space. This display is affected by the S and B display modes. The free space will be displayed in the most recently selected of these two display modes (display modes other than S and B do not affect this display.) If S has been selected the free space will be displayed in bytes. If B has been selected the free space will be displayed in blocks.

SELECTING ITEMS:

To select items to be affected by the action gadgets you simply click on file or directory names in a directory area. You can click on an item a second time to deselect it or click on the Deselect All gadget to deselect all of the items in a directory area. There is an Deselect All gadget for each directory area.

You can also select every item in a directory area by clicking the Select All Gadget. You can select a group of filenames by clicking and holding the left mouse button and dragging the pointer over several names. Then, when you release the button, all of the names you dragged over will be selected (or deselected, if previously selected). With the current version of ClickDOS II the directory area will NOT scroll if you drag past the last visible name.

Because of this, the most items you can select in this way is the visible area of names (15 in non-interlace or 40 in interlace mode), but it is still better than click, click, click, click....

V2.00 Finally, a very quick way to select items is to use the Wild Card Select feature of ClickDOS II. To use Wild Card Selecting click the Select All gadget of one of the directory areas using the RIGHT mouse button. A requester will appear asking you to enter a wild card.

Type in any valid AmigaDOS or ARP wild card and press Enter. ClickDOS II will select everything, including directories, in the directory area that matches the wild card.

The counterpart to Wild Card Selections is Wild Card Deselections. To use Wild Card Deselecting you first must have selected something. Then, you click ***the RIGHT*** mouse button over the Deselect All Gadget. A requester will appear asking you to enter a wild card. Type in any valid AmigaDOS or ARP wild card and press *Enter*. ClickDOS will deselect everything in the directory area that matches the wild card. This is very handy for copying everything in a directory EXCEPT certain items (like .info files).

After ClickDOS II has completed an action you requested with any action gadget, it will automatically deselect all files in BOTH directory areas. If you wish to do a second action on these same files, you can click the Select All gadget and it will reselect them for you PROVIDED the previous action did not cause ClickDOS II to re-read the directory in that area. Clicking on the Select All gadget twice will select ALL of the files in the appropriate directory area.

IMPORTANT NOTE: You will not be able to see selected files which have scrolled out of the directory area. BEWARE OF THIS! It is easy to forget about these files and inadvertently affect

them with an action. ClickDOS II helps out with this by deselecting all files after each action is complete.

ESTIMATING DISK USAGE: *(ClickDOS was developed with the older FFS, before large disk sizes were available, and will report negative numbers)*

As you select and deselect items, ClickDOS II keeps a running total of bytes or blocks selected for each directory area. This value is always displayed in the Amount Selected/Free Space display above each directory area. It is displayed as S:XXXXXXXXXX where XXXXXXXXXXXX is the number of bytes or blocks selected depending on the most recently selected of the two display modes: S or B (other display modes have no effect on this display).

The “Amount Selected” display is very handy for determining if all of the items you selected will fit on the disk to which you are copying them. Note that the blocks display mode is better for determining if you have enough free space because it takes into account the blocks for directory and file headers as well as file size. The Bytes display does not account for this space and thus gives Amount Selected values which are too low.

Unfortunately, AmigaDOS has no way of determining how large an entire directory of files is short of reading the entire directory, which, as everyone knows, is SLOW! Therefore, the contents of selected directories are not automatically included in the Amount Selected value. You can, however, have ClickDOS II scan all selected directories and add up all the space they occupy. You do this by clicking on the Amount Selected/Free Space Gadget. The total bytes or blocks occupied by everything you have selected in the directory area will be momentarily displayed in the message bar.

The Amount Selected display will NOT change to include the size of selected directories' contents. It will still only display total selected file size (plus 1 block for each selected directory if the Block display mode is active).

SOURCE AND DESTINATION DIRECTORY AREAS:

Actions such as COPY and MOVE require both a source directory and a destination directory. ClickDOS II makes determining which directory area is the source and which is the destination very simple - if you select a file or subdirectory in a directory area, that area automatically becomes the source and the opposite directory area becomes the destination. Therefore, to copy a file from one directory to another, you simply bring up the directory which contains the file in one directory area and the directory where you wish to copy the file in the other area. Which area is which is not important because you tell ClickDOS II which is the source by selecting the file.

To copy a file to the same directory but with a different name, you have ClickDOS II bring up the same directory in both areas. This is easy with the Swap Gadgets. One of the unique things about ClickDOS II is that you can select files in both areas at the same time and ClickDOS II will know which is the source and which is the destination for each file. You could, for example, pull up two directories, select files in both, then select MOVE and ClickDOS II would swap the files between the directories.

ClickDOS II always processes files from top to bottom, in the Left directory area first then the Right. It does NOT process them in the order they were selected as some directory utilities do.

WHAT ClickDOS II CAN DO FOR YOU:

This section describes ClickDOS II's built-in action gadgets.

MAKEDIR -

This is used to create a new subdirectory in one of the directory areas. When you click on this gadget ClickDOS II will present a requester asking you for the name of the new directory.

V2.00 If only one directory area contains a directory then that is where the new directory will be created. If both areas contain a directory then there will be three gadgets in the MakeDir requester.

Click on the LEFT gadget if you want ClickDOS II to create the new directory in the left directory area.

Click on the RIGHT gadget if you want ClickDOS II to create the directory in the right directory area.

Click on the CANCEL gadget if you have changed your mind.

V2.00 Normally, the MakeDir Requester will disappear after you create one directory. However, you can specify in the ClickDOS-Startup file that the MakeDir Requester is to stay active until the CANCEL gadget is selected. In this mode the LEFT and RIGHT gadgets (or pressing Return if only one directory area is occupied) will create the directory, then immediately redisplay the MakeDir Requester. This allows you to easily create several directories at one time. See the section on the ClickDOS-Startup file below.

RENAME -

Rename will let you change the name of one or more files. To use this option, select the files to rename then click on this gadget. ClickDOS II will ask you for a new name for each of the files you selected. At each requester, type in a new name or select CANCEL if you want to stop the rename process.

NOTE: Selecting CANCEL will only stop the rename process, it will NOT undo the files that were already renamed.

To make renaming easier, ClickDOS II will fill the requester gadget with the old name of the file. This allows you to edit the old name making renaming quicker. If you are completely changing the name, a quick Right Amiga-X will zap the old name out of the requester.

Rename will NOT let you rename across directories on the same disk as AmigaDOS will let you do. You must use the MOVE action to do this.

COPY-

Copy duplicates all selected files. The duplicate is either placed in another directory and is given the same name as the source file or it is placed in the same directory and given a different name. Where the duplicate is placed depends on the destination directory area.

To copy files, bring up the directory you wish to copy files FROM in one directory area and the directory you wish to copy files TO in the other directory area (which directory area is which, is not important.) Select the files you wish to copy and click the COPY gadget.

If you wish to copy files to the same directory but with different names, bring up the same directory in both areas, select the files you wish to copy in one of the areas and click COPY. ClickDOS II will know that you are copying files to the same directory and will ask you for a new name for the new copy of each file.

A very powerful feature of the copy action is that it can copy an entire directory! If you click on a directory name and then click COPY, ClickDOS II will create this directory in the destination directory and then copy every file AND SUBDIRECTORY and every file in every nested subdirectory in the source directory to the destination (phew!).

For example, you could copy an entire directory of fonts by simply clicking on the directory and then clicking COPY! Try it, I think you'll like it!

NOTE: If an error occurs while ClickDOS II is copying a file, ClickDOS II WILL remove the damaged destination file. However, if you are copying an entire directory, the files which had already been copied before the error occurred will NOT be removed from the destination. Only the single damaged file will be removed.

An error causes the COPY action to halt. It will not copy the remaining selected files after an error has occurred. You will need to look at the destination disk and see which files made it and which ones didn't. The MOVE action and the DELETE action also behave in this manner.

THE CLONE FEATURE:

Normally when AmigaDOS copies a file, the protection bits, comment and creation date are not copied along with the file. With the 1.3 release of AmigaDOS a new option was added to the COPY command called CLONE which would copy this supplementary data.

ClickDOS II also has a CLONE option with its COPY's and MOVE's. However, ClickDOS II's clone *only affects the protection bits and comment*. The creation date is not retained. In the center of the ClickDOS II window is a gadget called CLONE. You can click on this gadget to toggle the CLONE option on and off. The CLONE option is on when the word CLONE is surrounded by a highlight box.

You can set whether the CLONE option defaults to on or off by using the CLONE command in the ClickDOS-Startup file discussed below.

V2.00 THE OVERWRITE CONFIRM FEATURE:

ClickDOS II helps prevent data loss by providing an **Overwrite Confirm** feature. In the center of the ClickDOS II window just above the CLONE gadget is a gadget labeled OWC. This is the Overwrite Confirm gadget. It functions much like the CLONE gadget. Clicking the gadget toggles Overwrite Confirm on and off.

Overwrite Confirm is on when the gadget is surrounded by a highlight box. When Overwrite Confirm is on COPY, MOVE and MAKEDIR will check the destination directory for the existence of an item of the same name as the source item before they proceed with the operation. If an item is found in the destination a requester will appear which allows you to do one of three things:

1. You can type in a new name for the **SOURCE** item. This leaves the item in the destinate area intact.

2. You can click the **OVER** gadget to tell ClickDOS II to go ahead and overwrite the item in the destination.

3. You can click the **ALL** gadget. This tells ClickDOS II to perform the same action as the OVER gadget (i.e. overwrite the destination) but it also tells ClickDOS to temporarily turn off Overwrite Confirm for the remainder of the current action. For example, if you are copying one directory to another and you know most of the files are going to already be on the destination, it would be very tedious to have to click the OVER gadget for every file. The ALL gadget allows you to tell ClickDOS, "I know they're already there, but don't bother me about it just this once."

The Overwrite Confirm feature will work as usual during the next COPY, MOVE or MAKEDIR action. Even if you select the OVER gadget to tell ClickDOS II to copy over or make a new directory over and old directory the action will fail if the old directory is not empty. You can only overwrite empty directories.

You can set whether the Overwrite Confirm feature defaults to on or off by using the OWCFM command in the ClickDOS-Startup file discussed below.

MOVE-

The Move option works almost identically to Copy with one important difference - Move DELETES the SOURCE files or directory when it is finished copying them while Copy does not.

When Move is used to transfer files from one directory to another on the same disk it does so by renaming the files rather than actually making new copies. This is faster and a more efficient use of disk space.

DELETE-

Delete will permanently erase every file you have selected. Watch for those files which have scrolled out of the visible area that you forgot you selected!!

Delete will also delete an ENTIRE directory and every single thing in it!! BE WARNED! This is a very powerful but potentially VERY DANGEROUS feature. You should be sure you truly want to delete EVERYTHING in the directory you selected!!!!

V2.00 ClickDOS II provides four levels of data protection with the DELETE action. You set the level you wish to use in the ClickDOS-Startup file discussed below.

Level 0 - ClickDOS II will NOT ask for confirmation of any type. USE AT YOUR OWN RISK!!!!

Level 1 - ClickDOS II will NOT ask you to confirm the deletion of files but will ask you to confirm the deletion of directories.

Note that even if confirmation is ON ClickDOS II will ask you to confirm directory deletions only ONCE for each directory area each time you select DELETE. That is, if you have ten directories selected in one directory area ClickDOS II will not ask you to confirm the deletion of each directory, only the first one. This is to make deletions of multiple directories quicker but means you still need to be a little careful.

Level 2 - This level has the same features as level 1 but asks you to confirm the deletion of ANY item. Again, ClickDOS II does NOT ask you to confirm each item. It only asks once for each directory area.

Level 3 - This level combines levels one and two. It asks you to confirm the deletion of all items and if any items are directories it also asks you to confirm that you want to delete an entire directory.

A Lesson:

Confirmation of deletions was added to ClickDOS II because I, Gary Yates the author of ClickDOS II, accidentally deleted ALL of the sources of ClickDOS II using ClickDOS II. How?

I was getting ready to back up a full day's work on ClickDOS II. I selected all of the source files in one directory area then remembered I needed to delete some files on the disk I use for backups. Forgetting I had already selected the ClickDOS files, I promptly pulled up the backup disk in the other directory area, selected the files, clicked DELETE and WHAM!! I sat dumbfounded.

ClickDOS II now has deletion confirmations. I stay at level three. I recommend you do the same.

NEWCLI-

This gadget will bring up a NewCLI (or NewShell) window for you to use. This is a true CLI window and will hang around until you give the EndCLI command. The SHELL option in the **S:ClickDOS-Startup** file lets you tell ClickDOS what command to use to start the CLI (NewCLI, NewSHELL, AShell, etc.) and the default window size.

SHOW-

This option will show any IFF ILBM picture or brush. To show a picture select the picture's file name and then click the SHOW gadget. Show will display the picture for about five seconds then return to the ClickDOS II window. You can do a slide show type viewing by selecting several pictures. They will be shown one at a time. Between each picture you will see ClickDOS II flip back to its window.

This occurs for two reasons: one, so ClickDOS II can tell you the name of the next picture; two, this way only one screen has to be allocated in memory at any given time so show is more memory efficient.

Please keep in mind this option was intended to give you a way to review your files for deletion, copying, etc. and not as a true presentation program. It is useful but has no frills.

If you wish to view a picture for longer than five seconds, press the ESC key while the picture is on the screen. This will turn off the timing. Then, when you are finished viewing the picture, press ESC or the left Mouse Button to go to the next picture. Timing is turned back on for each picture.

To jump to the next picture before the five seconds has elapsed press the **LEFT Mouse Button**. To jump out of the slide show completely (i.e. return to ClickDOS II) press Q (for Quit).

If color cycling is defined in the IFF file, you can turn on color cycling by pressing the TAB key. Note that many pictures may have color cycling defined in the IFF file even though the picture was not intended to be color cycled.

For example, Deluxe Paint puts color cycles into every picture regardless of whether the artist used color cycling or not. For such pictures, turning on color cycling can have interesting but unusual results.

Because of such problems ClickDOS II does NOT automatically turn on color cycling; you must do it with TAB. To turn off color cycling press TAB a second time and ClickDOS II will restore the colors to their original state. ClickDOS II turns OFF color cycling as it loads each picture.

As mentioned above Show can display brushes. It can also display overly large bitmaps BUT only as much of the upper left corner as will fit on the screen. Show supports both HAM (Hold And Modify) and Extra Half Bright modes. Show is, however, a very simply IFF reader. It will NOT find ILBM hunks which are inside of other hunks, for example, ANIM hunks.

V2.00 While SHOW does not have a built-in ANIM viewer (a built-in ANIM viewer would make ClickDOS II too large), if you specify an ANIM viewer in the ClickDOS-Startup file, SHOW will automatically execute the ANIM viewer when it reads an ANIM IFF file. See the ClickDOS-Startup section below. See also the section ICONIFYING ClickDOS II below.

TYPE -

This option lets you view any text or binary file in ASCII format. It does no formatting of its own (i.e. no word wrap) I did not implement this because so many README and DOC files (the most common use of type) are pre-formatted. ClickDOS II trying to reformat these already formatted files just makes a mess!

V2.00 However, as of version 2.00 ClickDOS II does recognize tab characters and handles them as eight column tab stops just as the AmigaDOS TYPE command does.

Type opens a separate borderless window to show you files. It will show you several files one after the other if you select multiple files.

Type converts all non-printable characters in a file to spaces when it displays them. This makes it easy to see text which is in a binary file. You can use the following commands to scroll through the text:

SPACE OR LEFT MOUSE BUTTON - scrolls forward through the text by one full screen

RETURN - scrolls forward by one line

RIGHT MOUSE BUTTON - scrolls backward by one full screen. Same as B command below.

In addition to these commands the title bar of the TYPE window displays other commands available while typing a file. These commands are:

b or B - scrolls Backward by what ClickDOS II can "best guess" is one screen. ClickDOS II does not actually take the time to scan backwards in the file and figure out how to back up exactly one screen full of text. Instead, it simply guesses how far back to go based on the number of characters on the current screen. ClickDOS II will tell you how many characters it went backward in the file.

g or G - allows you to skip directly to a certain percentage of the file. You will be presented with a requester asking for the percentage of the file you wish to skip. For example, if you want to jump to the end of the file enter "100" for 100 percent.

If you wish to jump to the middle of the file enter 50 for 50 percent, etc. ClickDOS II does not support jumping to a specific line number.

t or T - takes you to the top of the file.

e or E - takes you to the end of the file.

V2.00 m or M - (Modify) starts the editor which you have specified in the ClickDOS-startup file passing it the name of the file currently being typed. TYPE will immediately display the next selected file (if there is one). When you exit your editor the file in the TYPE window will not be the file you just edited.

s and S - perform simple NON-CASE sensitive searches. When you do a search ClickDOS II will present you with a requester into which you type the string of characters for which you wish to search.

The search begins at your current location in the file, NOT at the top. If you wish to search the entire file, you need to press **t or T** first, to go to the top.

When ClickDOS II finds the search string in the file it will reposition to the line containing the search string and display the next screen of text following the line. ClickDOS II (**V2.00**) will place an asterisk on the top line of the type window showing the beginning of the search string ClickDOS II found in the line. If the search string is not found, ClickDOS II will reposition to the bottom of the file and display:

---{ NOT FOUND }--- on the top line of the window.

The **lowercase s and capital S** search options do not behave exactly the same. Capital S will always force ClickDOS II to give you a search requester so that you can enter a search string. Lowercase s, on the other hand, will only present a requester if you have not done a search previously (i.e. the search buffer is empty.) Therefore, you can use capital S to do new searches and lowercase s to repeat the last search (i.e. search for the next occurrence of the same string.)

ClickDOS II allows you to search for non-printable characters such as linefeeds and tabs. You do this by entering a letter representing the non-printable character preceded by the ARP escape code. By default the ARP escape code is an asterisk (*) which is also the escape code used by AmigaDOS. Please note that this is different from versions of ClickDOS II prior to **V2.00**. These previous versions, which did not use ARP, used the backslash as the escape code. If you like the backslash better, you can use the ARP SET command to change the escape code to another character such as the backslash (\). See the ARP documentation.

The following are the letter symbols for the non-printable characters:

New Line(line feed) - *n

Carriage Return- *r

Bell- *g

Form Feed- *f

Tab- *t

In addition you can use *xnn, where nn is a hexadecimal ASCII value, to search for ANY ASCII code including non-printable characters.

Because the asterisk is used to flag these special characters, in order to search for an asterisk you must type two asterisks (**). The first one acts as an escape code the second tells ClickDOS II to search for the asterisk. To search for two asterisks you type four (****), etc. Note, if you change the escape code to something other than the asterisk then you do not have to specify two asterisks to search for an asterisk; instead, you will have to type two of the characters you chose as the escape code to search for that character.

V2.00 P AND P - This command allows you to print to your printer from within the TYPE action. Typing a lowercase p prints what is currently in the TYPE window.

This is extremely handy for printing only small sections of a very large file. It is similar to the Print-Screen function available on MS-DOS (shudder!) machines. This option creates a temporary file in RAM: so that you do not have to wait for the printing to finish before you continue to work with ClickDOS II.

The SPOOL program (see below) will automatically delete this temporary file when it has finished printing it. See Appendix B for more information on ClickDOS II's temporary files.

Typing a **capital P** will print the entire file you are currently typing. This is equivalent to selecting the file and then selecting the PRINT action. This feature was added to TYPE for convenience.

Both of the TYPE Print options use the SPOOL supplementary program. See Appendix A for more information on Spool.

ESC - Ends viewing the currently displayed file and jumps to the next selected file. If the current file is the last file selected, then ESC has the same effect as Q

Q - Ends viewing files and returns to the ClickDOS II window ignoring the remaining selected files.

Also see the section **ICONIFYING** ClickDOS II below.

PRINT -

This action allows you to send ASCII files directly to your printer (if you have one). You can select several files then click on PRINT and ClickDOS II will spool all of the files to the printer. To do its print spooling ClickDOS II uses a supplementary program called Spool. See Appendix A for more information on Spool.

RUN -

The RUN option allows you to execute a program or script file from within ClickDOS II. There are two ways you can use run: one, you can select a program name in the directory area and then click on RUN to execute that program or script; two, you can simply click on RUN and ClickDOS II will open a requester asking you for the name of the program or script to execute.

In either case ClickDOS II will present you with a requester asking for optional arguments to pass to the program or script. Any arguments you type will be passed to the program or script just as if you had typed them after the name of the program or script at a CLI. Just like the AmigaDOS shell, ClickDOS II determines whether you are attempting to start a program or

script by looking at the script flag of the file. If the script flag is set ClickDOS assumes the file is a script. Otherwise, it assumes the file is a program.

If ClickDOS II determines that the file is a program, (i.e. the script bit NOT set) then the argument requester will contain three gadgets. Regardless of whether you type in any arguments, you must click one of these gadgets before ClickDOS II will execute the program. These gadgets are:

STD IO - These stands for Standard Input/Output. If you click on this gadget ClickDOS II will open a normal console window, much like a CLI window, for the program to use for its input and output.

CANCEL - Click this to cancel running the program.

NO IO - This starts the program as a background process similar to typing the following at a CLI:

RUN <NULL: >NULL: program <arguments>

If ClickDOS II determines that you are attempting to execute a script file, only one gadget, CANCEL, will appear in the requester. Regardless of whether you type in any arguments, pressing return will cause ClickDOS II to open a standard output window and execute the script.

ClickDOS II will not execute scripts as background processes. Clicking CANCEL without pressing enter will cancel execution of the script.

V2.00 Once you select STD IO or NO IO (or pressed return if you are executing a script) ClickDOS II creates a temporary file in RAM and starts the supplementary program called SPAWN.

SPAWN executes the command which is in the temporary file. Use of the SPAWN program allows ClickDOS II to start subprocesses but not have to wait for them to finish executing like the CLI does. As soon as SPAWN is started control returns to ClickDOS II.

Limitations of RUN:

AmigaDOS has very POOR support for having one program (such as ClickDOS II) start another program. This is one of the major reasons ClickDOS II **V2.00** requires the ARP library and uses the SPAWN program. While these things have improved the reliability of RUN, there are still some limitations to what RUN can do.

V2.00 * Even though ARP helps, some programs which require a standard input stream may have problems. Using ARP did make DiskCopy and Format work correctly but the AmigaDOS Ask command still will not work.

You should be cautious of RUNning programs which will require input from the CLI; Programs which open their own window should work perfectly fine. Trying to RUN a program which uses standard input can have one of three effects; one, it will work (GOOD!!); two, it will hang the program (BAD!!!); three, the program will detect that there is no standard input and report an error (OKAY) or continue with some default value (SOMETIMES BAD SOMETIMES GOOD). You need to experiment to see how programs behave.

* Using ARP did not seem to help with executing scripts. Therefore, ClickDOS II **V2.00** executes scripts the same way as previous versions. This means when you execute a script, the script will get NO standard input, only a standard output. This means scripts which use

commands such as ASK, or others which require input, will fail. Scripts should execute fine unless they require standard input, or start a program which requires standard input.

DO NOT run Diskcopy or Format from a script!

* Because the AmigaDOS NIL: device is not a true device, it sometimes causes problems.

Therefore, ClickDOS II's RUN action, and User Defined gadgets (see below), require use of the NULL: device. NULL: is a public domain device driver which you mount, with the MOUNT command, in your startup-sequence. Use of NULL: makes the RUN and User Defined gadgets behave in a much more reliable manner. In fact, I highly recommend you use NULL: for all operations where you would normally use NIL:. Thus, NULL: is much safer, but doesn't work in KS 2.04 and above.

I know all of this may not make sense to some people so here are some basic rules to follow when using RUN:

1. Except for Ask, DiskCopy and Format, none of the AmigaDOS commands require standard input unless you specify a '?' as their argument. So unless you use the '?' argument all of the AmigaDOS commands (or their ARP equivalents) should work correctly. ClickDOS II's use of the ARP library also allows DiskCopy and Format to be used correctly, but Ask will still not work.

WARNING!!! Diskcopy and Format will start without waiting for you to put in the correct disks if you select NO IO as their input/output option. Always select STD IO when using DiskCopy or Format from RUN! You can very easily format or copy over the wrong disk if you don't. You have been warned!!!

2. Do not RUN a program unless you have started it before, from a CLI or from Workbench, and you know how it behaves.

3. If a program opens its own window and does everything from that window, then it is probably safe to use with RUN with either the STD IO or the NO IO options.

4. If a program only does output to a CLI window but no input (i.e. it never asks you to type in anything) then it is probably safe to use with RUN with the STD IO option.

5. If you know a program does input from a CLI, such as ASK (which asks you to type in Y or N) then it may NOT be safe to use with RUN! Test the program carefully and be prepared for a GURU.

6. If a program will run from Workbench by double clicking its icon, it may be safe to use with RUN with either STD IO or NO IO. Do some testing.

I hope that Commodore will enhance the ability of AmigaDOS to handle child processes in the 1.4 release. In my opinion this is by far the weakest part of AmigaDOS.

PROTECT -

The PROTECT action lets you change the protection flags of files and directories. ClickDOS II supports the AmigaDOS 1.3 protection flags (s)cript, (a)rchive and (p)ure. To change the protection status of an item, select the files or directories whose protection flags you wish to change then select the PROTECT gadget. A requester will appear showing you the protection flags of the first file. ClickDOS II will show you, in the message bar, the name of the file for which you are changing the protection flags. In the requester are eight gadgets, one for each of the currently supported protection flags. The gadgets correspond to the protection flags as follows from left to right:

(H)idden

(S)cript

(P)ure
(A)rchive
(R)eadable
(W)ritable
(E)xecutable
(D)eletable

Note, with 1.3.2 release of AmigaDOS the AmigaDOS PROTECT command will set and clear the (H)idden flag. However, AmigaDOS seems to ignore this flag at this time. ClickDOS II will also set and clear this bit, but also ignores it. See your AmigaDOS manual for the meaning of the other protection bits.

The text in each gadget of the protect requester tells you if the protection flag is on or off. If the gadget contains the first letter of the name of the protection flag, then the flag is on. If the gadget contains a dash (-) then the flag is off. To change a protection flag, click on the gadget which corresponds to the flag you wish to change. This will toggle the protection flag on if it was off and vice-versa. Once you have the protection flags set the way you want for the indicated file, you must then select one of the other three gadgets in the requester:

OK - If you want to use these protection flags only for the file indicated in the message bar then select this gadget. ClickDOS II will set the flags for that file then present another requester for the next selected file.

ALL - If you want to use these protection flags for all of the remaining selected files then click ALL. ClickDOS II will set the protection flags in the current requester for all of the remaining selected files.

CANCEL - If you changed your mind then select CANCEL. Note that if you had already selected OK for the protection flags set on several previous files ClickDOS II will NOT cancel the changes to those files. It will only cancel the current requester and end the PROTECT action for the remaining selected files.

COMMENT -

Use this action to set an AmigaDOS comment for a file or directory. To add or change the comment of an item, select the files or directories whose comment you wish to add/change then select the COMMENT gadget. A text requester will appear showing you the current comment, if any. ClickDOS II will show you, in the message bar, the name of the file for which you are adding/editing a comment.

Use the text gadget to add a new comment or edit the existing one. When you have finished editing the comment, you must then select one of the other three gadgets in the requester:

OK - this sets the comment in the text gadget for the indicated file, then gives you a new requester for the next selected file.

ALL - this sets the comment in the text gadget for the indicated file and for all of the remaining selected files.

CANCEL - cancels the COMMENT action.

EDIT -

This action allows you to edit files directly from within ClickDOS II. The EDIT action starts the editor which you have specified in the ClickDOS-Startup file. ClickDOS II passes this editor the name of the FIRST file you have selected in a directory area (remember ClickDOS II searches

from top to bottom in the left directory area and then the right). If more than one file is selected only the first file is edited.

All other files are ignored. If no files are selected then you will be presented with a requester asking you for the name of the file to edit. Also see the section below about the ClickDOS-Startup file.

V2.00 USER-DEFINED GADGETS:

There are six action gadgets in the ClickDOS II window which can be user defined to execute external programs. These allow you to customize ClickDOS II so it performs actions of your choice. How to setup user defined gadgets is explained in the section covering the ClickDOS-Startup file.

ICONIFYING ClickDOS:

You will notice (perhaps with some reservation) that the ClickDOS II window does not have a size gadget. This is true, you cannot size the window. There is however an ICONIFY system gadget in the title bar of the window, beside the front back gadget. This gadget, when clicked, will shrink the ClickDOS II window down to a tiny size and place it in the title bar of the Workbench screen. Clicking on this gadget again will return the window to its normal size.

This is called iconifying. When its window is iconified ClickDOS II uses much less chip memory and is out of the way. ClickDOS II takes up about 72K of FAST memory and 2K of CHIP memory when it is iconified, and you have expansion memory. ClickDOS II also "goes to sleep" when it is iconified so it takes up almost no CPU time (only a small amount of time to update the memory display). **IT IS VERY HANDY TO LEAVE CLICKDOS II ICONIFIED AND AVAILABLE ALL THE TIME ON YOUR WORKBENCH**, if you have the memory. If you would like to start ClickDOS II in its iconified state, you can use the -i argument when you start it from a CLI. This is useful if you would like to start ClickDOS II from your StartUp-Sequence.

If you do not like the iconified window where ClickDOS II places it, you can drag it to a new location and from then until you end ClickDOS II the iconified window will be placed there. *Also, you can specify absolute screen coordinates* in the ClickDOS-Startup file and these will become the default location of the iconified window.

You can also iconify ClickDOS II while the Type or Show windows are displayed. However, while either of these windows is displayed the iconify gadget is disabled. To iconify ClickDOS II while the Type window is displayed, click the **RIGHT Mouse button** in the title bar of the Type window.

The Type window will close and the ClickDOS II window will shrink. Click the iconify gadget to un-iconify ClickDOS II. The Type window will re-open. To iconify ClickDOS II while the Show window is displayed simply click the RIGHT mouse button anywhere on the window. The Show window (actually a separate screen) will be pushed to the background, and the ClickDOS II window will shrink.

To un- iconify ClickDOS II click on the iconify gadget. The ClickDOS II window will un-iconify, and the Show window will come back to the front. While ClickDOS II is iconified with the Show window open, the normal Show timer is turned off. Timing will begin again when

ClickDOS II is un-iconified. Also note that if you iconify ClickDOS II while showing a picture the Show window remains open and is consuming a large amount of memory!

The “ClickDOS-Startup” FILE

When ClickDOS II starts executing it will try to read a file called **ClickDOS-Startup** in the directory where S: is currently assigned. Please note that this has changed from ClickDOS V2.06 which read a file called .DOSRC in the T directory. The formats of the .DOSRC file and the ClickDOS-StartUp file are quite different.

The ClickDOS-Startup file is an ASCII text file which contains commands telling ClickDOS II how to configure itself. Each line of the **ClickDOS-Startup** file has the following format:
command=value;

All of the startup file commands are **five letters long** and must be **in all capital letters**. Each command must be followed by an equals sign and then an appropriate value for that command.

There can be no spaces around the equals sign. Each line must end with a semi-colon (;). You may place a comment on the same line after the semi-colon if you wish.

I made ClickDOS II very strict with its format of the startup file to keep the size of the initialization code very small. The initialization code of a program gets executed once when the program starts and then is just "dead weight" for the remainder of the time the program is in memory. Keeping the size of ClickDOS II small makes it more useful to more people, namely those with limited memory.

Here are the commands you can specify in the ClickDOS-Startup file to tailor ClickDOS II to your needs and tastes:

VOLMS -

This command allows you to set the device names which you want to appear in ClickDOS II's device gadgets. It must be followed by a string of 28 characters which define the devices, for example:

VOLMS=RAM:RAD:DF0:DF1:DF2:C: DH0;;

No device name may be over four characters. If a device name is less than four characters, then it must be padded on the right with spaces (see C: above).

Note versions of ClickDOS II prior to **V2.00** had a device string of 32 characters because ClickDOS II had eight device gadgets at that time. As of **V2.00** ClickDOS II has only seven device gadgets due to the addition of the volume request gadget.

If you specify a VOLMS command with a device string longer than 28 characters, any devices listed after the 28th character will be ignored.

IXPOS -

This option lets you specify the default X coordinate location of ClickDOS II's iconified window. It must be followed by a number for the location of the left-hand edge of the window, for example:

IXPOS=120;

This would put the left edge of ClickDOS II's iconified window at the 120th pixel on the workbench screen. Legal values for IXPOS are 0 to 200.

IYPOS -

This option allows you to set the default Y coordinate location of ClickDOS II's iconified window. It must be followed by a number for the location of the top edge of the window, for example:

IYPOS=0;

This would put the top edge of ClickDOS II's iconified window at the very top of the Workbench screen. Legal values for IYPOS are 0 to 190 for a non-interlace Workbench screen, and 0 to 390 for an interlace Workbench screen.

NOTE: If you are going to specify an IYPOS value higher than 190 then the command **ILACE=Y;** must be specified before the IYPOS command in the startup file.

SPOOL -

This command allows you to tell ClickDOS II where it can find the Spool program. The default location is (C:SPOOL). It also allows you to tell ClickDOS II that you have renamed the Spool Program. You may want to rename Spool, for example, if you already have a program called Spool and you wish to keep both in your C directory. Here are two examples of the SPOOL command, one which tells ClickDOS II you have renamed Spool, and one which tells ClickDOS II that Spool is somewhere other than the C: directory:

SPOOL=C:Print It; SPOOL=DH0:ClickDOS_Programs/Spool;

If you wish to both rename Spool and move it to another directory, you would use a SPOOL command something like this:

SPOOL=DH0:ClickDOS_Programs/Print_It;

WARNING: ClickDOS II only allocates 99 bytes to store the location of Spool. You need to be sure that the string which follows your SPOOL command is not more than 99 characters long. ClickDOS II DOES NO length checking.

V2.00 SPAWN -

Similar to SPOOL, this command allows you to tell ClickDOS II where it can find the Spawn program and if you have renamed it. The default location is (C:SPAWN).

WARNING: ClickDOS II only allocates 99 bytes to store the location of Spawn. You need to be sure that the string which follows your SPAWN command is not more than 99 characters long. ClickDOS II DOES NO length checking.

CLONE -

This option allows you to tell ClickDOS II if you want the CLONE function to be on or off by default. If you want cloning on, then use:

CLONE=Y;

If you want cloning off, then use:

CLONE=N;

DSPLY -

This option allows you to tell ClickDOS II which display mode you want the directory areas to use by default. You follow the DSPLY command with one of the following letters:

S - display file size in bytes

D - display file creation date

T - display file creation time

P - display protection bits

C - show if a file has a comment

N - show nothing but the file names

B - display file size in bytes

Display modes must be specified in capital letters.

For example:

DSPLY=D;

ILACE -

You use this option to tell ClickDOS II that you want to take advantage of the extra space available on an interlace Workbench screen. If you use an interlace Workbench screen and you want ClickDOS to expand to fill the entire screen use:

ILACE=Y;

If you use an interlace Workbench but DO NOT want ClickDOS II to expand to fill the screen use:

ILACE=N;

or leave this command out entirely. If you do not use an interlace Workbench screen then ClickDOS II will ignore this command.

SHELL -

You use this command to tell ClickDOS II which command to execute to start a new CLI rather than the default (C:NEWCLI), for example:

SHELL=C:NEWSHELL;

this would tell ClickDOS II to use the new AmigaDOS 1.3 NewShell command to start new CLI's. You can also follow this command with a valid console definition to tell ClickDOS II how to open the CLI window, for example:

SHELL=C:NEWSHELL CON:0/0/640/100/My_New_CLI;

See your AmigaDOS manual for more about console definitions. ClickDOS II allocates 99 characters to hold the Shell definition.

CONSL -

This command tells ClickDOS II what console definition to use to open the STD IO console window for the RUN and EDIT actions. You follow this command with any valid console definition, for example:

CONSL=CON:0/0/640/100/My_spawned_task;

See your AmigaDOS manual for more about console definitions. ClickDOS II allocates 99 characters to hold the Console definition.

V2.00 CNFRM -

This command allows you to set ClickDOS II delete confirmation level. By default ClickDOS II will NOT warn you before you delete an item. To set the delete confirmation level follow the CNFRM command with a number. There are three confirmation levels:

0. No confirmation (the default)

1. Confirm directory deletions

2. Confirm all deletions

3. Confirm all deletions and also confirm directory deletions

See the section on the DELETE action for more information on the delete confirmation levels.

STIME -

This command allows you to set the delay time which the SHOW action uses while displaying a picture. By default the SHOW action pauses five seconds for each picture. You can use this command to make this delay longer or shorter. For example:

STIME=10;

would set the SHOW delay to ten seconds. The value for STIME must be between 1 and 90 seconds.

SBUFF -

This command lets you change the default size of the spool buffer which ClickDOS II tells the Spool program to allocate. See the description of the Spool program in appendix A for more information about the spool buffer. To change the size of the spool buffer uses the following command in the ClickDOS-Startup file:

SBUFF=XXXXX;

where XXXXX is the size of the spool buffer in bytes. The size of the spool buffer which you specify must be at least 1K (1024) and must be a multiple of 1024 (i.e. 1024 must divide into the value evenly.) For example:

SBUFF=10240;

would set the spool buffer to 10K. ClickDOS II allocates 9 characters to hold the Spool buffer size.

V2.00 EDITR -

Note in a previous edition of this documentation this command was mistakenly called EEDIT. You use this option to specify the editor which you would like the EDIT action to use. You need to include the entire path to the editor program. For example:

EDITR=C:ed;

This would cause ClickDOS II to use the ed editor which is in the directory where C: is assigned. ClickDOS II allocates 99 characters to store your editors path and name.

EDSTD -

You use this option to tell ClickDOS II if it should open a standard input/output window for the editor to use. If your editor needs a standard input/output window then use:

EDSTD=Y;

otherwise use:

EDSTD=N;

You will need to experiment with your editor to determine if it needs a standard input/output window. A good rule of thumb to use is if your editor opens its own window you will probably NOT need an input/output window.

V2.00 EDARG -

This command allows you to specify arguments to be passed to your editor when it is executed by ClickDOS II. ClickDOS II will use EDITR and EDARG to start your editor just as if you had started it from the CLI with the following command line:

<EDITR> <EDARG> <filename>

ClickDOS II allocates 99 characters to store these arguments.

V2.00 ANIMV -

You use this command to tell ClickDOS II what ANIM viewer it should use if you click the SHOW action and ClickDOS II recognizes that the IFF file is an ANIM file. You should include the entire path in this specification. For example:

ANIMV=DH0:Utilities/showANIM;

ClickDOS II always executes the ANIM viewer the same as if you selected the RUN action and use an IO mode of NO IO. See the sections for the SHOW and RUN actions for more information. ClickDOS II allocates 99 characters to store the name and path of your ANIM player.

V2.00 ANARG -

You need to use this command if your ANIM player needs arguments other than the name of the ANIM file. ClickDOS II will use ANIMV and ANARG to start your ANIM player just as if you had started it from the CLI with the following command line:

<ANIMV> <ANARG> <filename>

ClickDOS II allocates 99 characters to store these arguments.

V2.00 MSTAY -

If you would like the MakeDir Requester to continue to ask you for directories to create until you select CANCEL then use the following command:

MSTAY=Y;

The default is: **MSTAY=N;**

In which case the MakeDir Requester will ask you for one directory name then disappear. You will have to select the MakeDir action again to create another directory. See the section on the MakeDir action for more information.

V2.00 OWCFM -

If you would like to have Overwrite Confirm turned on then put the following command in your ClickDOS-Startup file:

OWCFM=Y;

otherwise use: **OWCFM=N;**

See the section on the COPY action for an explanation of the Overwrite Confirm feature of ClickDOS II.

V2.00 User Defined Gadget Definitions:

You set up a user-defined gadget in ClickDOS II by including an entry for the gadget in the ClickDOS-Startup file. You must have one entry for each gadget you are defining. Each entry has the following format:

USERx=<name>;

<console window definition>;

<command>;

<command arguments>;

The second through fourth lines must be indented six spaces as shown. The following is an explanation of each line of the entry. You should refer to the ClickDOS-Startup file which accompanied ClickDOS II. It has some good examples of user- defined gadget entries.

Line 1: The first line contains the command telling ClickDOS II which user-defined gadget this entry defines (USERx). You replace the x with the number of the gadget you are defining (1- 6). You follow this with an equals sign (=) then the name you want ClickDOS II to place in the gadget when it appears in the ClickDOS II window. The name CANNOT be more than 7 characters long. Example:

USER1=DiskCpy;

Line 2: The second line defines the console window needed by the command that will be executed when this gadget is selected. Console window definitions come in four forms:

@

@<console definition>

<console definition>

\$<console definition>

where <console definition> is any valid AmigaDOS console window definition. For example:

CON:0/0/640/200/My_Window

If you give the console window a name, as in 'My_Window' above, if the name contains spaces you must enclose the entire console window definition in quotes. For example:

"CON:0/0/640/200/My Window"

See the AmigaDOS manual for more information on console window definitions.

Each of the different types of console window definitions tells ClickDOS II a different way to execute the command associated with the user defined gadget.

@ (with no console definition)- tells ClickDOS II to execute the command as a background task. This is similar to using the RUN action and selecting NO IO.

<console definition> - An AmigaDOS console definition by itself tells ClickDOS II to use standard input/output mode for the command. This is similar to executing the command from the RUN action and selecting STD IO. You should use this mode when you need to see the output from the command because ClickDOS II will pause the console window when the command terminates.

@<console definition> - tells ClickDOS II to execute the command as a standard IO process with no wait after the command has finished executing. The console window will close immediately upon termination of the command. This type of IO mode is not available from the RUN action. It is useful for commands which:

- * require a standard input/output window
- * will take a fairly long time to execute
- * and you do not necessarily care about the output from the command

For example, the ClickDOS-Startup file that is distributed with ClickDOS II uses this mode for the DISKCOPY command. Here is an example of this type of console window definition:

@CON:0/0/640/200/DiskCopy

\$<console definition> - This tells ClickDOS II that the command is actually a script file rather than a program file. ClickDOS II will execute the script the same way the RUN action executes scripts. See the section on the RUN action for more information.

Here is an example of this type of console window definition:

\$CON:0/0/150/50/My_window

Line 3: The third line of the user-defined gadget entry contains the name of the program to execute and the path where ClickDOS II can find the file. For example:

dh0:c/assign

Line 4: Line four allows you to specify arguments for the command in line three. This is the most complex line of the user defined gadget entry and is what gives ClickDOS II's user-defined

gadgets their power and flexibility. You can do two things in the argument list specified in line four:

1. You can specify literal arguments for the command.
2. You can specify ClickDOS II replacement arguments (CRA's).

Literal arguments will be passed to the command exactly as they are specified. For example you could have the following user defined gadget entry:

```
USER1=Dir;  
CON:0/0/640/200/Dir;  
C:Dir;  
dh0;;
```

This would define a gadget to ClickDOS II called 'Dir' which executes the AmigaDOS DIR command in the C directory. When you clicked on the gadget, ClickDOS II would open a standard IO console window which is the full screen on a non-interlace Workbench. ClickDOS II would then execute the DIR command in that window passing it dh0: as a literal argument.

This means that every time you click the gadget you would get a directory of dh0: in a full-screen window. Now this probably does not sound very useful and your right. Literal arguments are not very flexible but they do serve a purpose when used in combination with ClickDOS II Replaceable Arguments (CRA's - pronounced "craze"). CRA's allow you to specify parts of the argument list which will be filled in by ClickDOS II before the command is executed. There are five types of CRA's. They are:

- %s - replace with String
- %d - replace with Device name
- %p - Pause
- %f - replace with First selected item
- %l - create List using selected items

These must be specified as percent signs followed by a **LOWERCASE** letter. The easiest way to explain the use of CRA's is by example. But first, here are the specific formats of the CRA's:

```
%s[default][prompt]  
%d[prompt]  
%p[prompt]  
%f[type][prompt]  
%l[type][~]
```

Let's look again at the 'Dir' example above but this time let's use CRA's to make it more useful. First, let's use the %s CRA. The %s, or string CRA, causes ClickDOS II to open a generic string input requester. Whatever you type into this requester is substituted for the %s CRA in the command's argument list before the command is executed. You follow the string CRA with two parameters:

1. A default value to put in the text gadget of the requester.
2. A prompt for the requester.

To use the %s CRA for the 'Dir' example you would change the fourth line of the user defined gadget entry to be:

```
%s[dh0:][Enter a directory to Dir:];
```

Now, every time you clicked the 'Dir' gadget ClickDOS II would ask you to enter a directory name in a string requester. It would give you a directory of the directory you enter. You could also use the %d CRA with the 'Dir' user defined gadget. The %d CRA causes ClickDOS II to

open its Device Requester. It inserts the device you select from the requester into the commands argument list. To use the %d CRA you would use the following fourth line:

```
%d[Select a device to Dir:];
```

Finally, let's show how you would use the %f CRA with the 'Dir' example. The %f CRA causes ClickDOS II to take the first file selected in the directory areas (remember ClickDOS II scans the areas from top to bottom left to right) and insert the filename and its full path into the argument list of the command.

%f must be followed by two parameters:

1. The "type" parameter which tells ClickDOS II whether it should use only the first selected file name (f), only the first selected directory name (d), or whatever the first selected item is regardless of its type (a).

2. The "prompt" parameter is only used if ClickDOS II does not find any selected items. In this case the %f CRA basically becomes a %s CRA and ClickDOS II will open a string requester with the specified prompt. For our 'Dir' example, it only makes sense to use the directory command on directory names so we will specify a d type:

```
%f[d][Enter a directory name]
```

With this entry, when the gadget is selected ClickDOS II would first check to see if a directory name is selected. If one is, the name will be substituted into the DIR command's argument list and you will get a listing of that directory's contents. If nothing is selected, you will get a string requester asking for a directory name. If a file is selected, ClickDOS II will give the error message:

Directory required!

Of course even with the flexibility the CRA's give our 'Dir' user-defined gadget, it still would not be very useful in ClickDOS II. To show a more useful command and a better example of how to use CRA's and literal arguments, let's do a 'Format' user-defined gadget.

The first step in setting up a user-defined gadget entry is to consider what you would type at the CLI to execute the command. To do a format from the CLI you would type something like this:

```
dh0:system/format DRIVE <device> NAME <name of disk> NOICONS
```

As you can see, if you think about the command, the drive to format and the name of the disk need to be CRA's so we can specify different drives and names from within ClickDOS II. All of the other arguments can be literals because they are the same for every format. We can use a %d CRA for the device since we need a device. And we can use a %s CRA for the name of the disk.

Another thing we need to consider when making a user-defined gadget entry is what type of console window definition we need to use for the command. FORMAT must have a standard IO window because it asks you to press return to start the format. If a program needs input from the CLI then it must have a standard input/output window.

However, we don't really need for the console window to pause when the format is done. We can satisfy both of these needs with the no-wait standard IO type of console window. Finally, format does not need a full-screen window because all we need to see is the line:

```
formatting: XX, XX to go...
```

Therefore, the following console window definition will work well:

```
@CON:0/0/150/50/Formatting...
```

The entire user-defined gadget entry would look like this:

```
USER1=Format;  
@CON:0/0/150/50/Format;  
dh0:system/format;  
DRIVE %d[Select a drive to format:] NAME %s[Empty][Enter a name for disk:] NOICONS;  
(Note: on your screen or when you print this file the last line of the entry above may be split and  
be on two lines. In the ClickDOS-Startup file lines should never be split.)  
Now, if you run ClickDOS II with this entry in the ClickDOS- Startup file you will get a  
'Format' user-defined gadget. If you click on this gadget ClickDOS II will display the device  
requester and ask you to select a drive to format. Once you have selected a drive a string  
requester will appear and ask you to enter a name for the disk. When you press Enter ClickDOS  
II will start the FORMAT command passing it the arguments you have selected.
```

A 150 by 50-pixel console window will appear on your screen and FORMAT will execute. It will ask you to press return to start the format (as usual). While the format is running you can continue to use ClickDOS II assuming, of course, that you have another drive besides the one doing the format. When the format finishes the console window will disappear.

As you can see, ClickDOS II's user defined gadgets can be very powerful and flexible. Now let's look at an even more powerful CRA the List CRA (%l).

The List CRA allows you to create user-defined gadgets that behave much like ClickDOS II's built-in action gadgets. That is, it allows your user-defined gadgets to operate on all of the selected files in one or both directory areas.

A perfect example to use for the list CRA is an archive program such as Zoo. With Zoo you need to specify a large number of files and have Zoo archive all of them into a single .zoo file. Here is the user defined gadget entry you would use to create a 'Zoo' gadget:

```
USER2=Zoo;  
CON:0/0/640/50/Zooing...;  
C:Zoo;  
-add %s[RAM:][Enter name of Zoo file:] %l[f][~];
```

There are some things you need to notice in this entry. Standard IO mode is used because we want the console window to stay open even when Zoo is finished. This allows us to see if any errors occurred during the zoo process. Also, you must be sure that you put the CRA's in the argument list in the same order that the command expects the arguments.

There are some extra considerations when using the List CRA:

- * Only one List CRA can be used in a user-defined gadget
- * If you use a %f you cannot use a %l CRA and vice-versa

Here is how ClickDOS II actually handles user-defined gadgets with List CRA's. We'll use the Zoo entry above as an example.

1. ClickDOS II processes all of the CRA's in the argument list except the List CRA. For example, in the case of Zoo, the %s CRA would be processed let's say you enter:

```
RAM:temp.zoo
```

So the argument list after the substitution would be:

```
-add RAM:temp.zoo %l[f][~]
```

2. Once the other CRA's have been substituted ClickDOS II opens a temporary file in RAM: and writes to it the full command specification once for each selected item in the directory areas substituting the name of each item for the List CRA. So if we say you have the following files selected:

RAM:myprog1

RAM:myprog2

RAM:myprog3

for the 'Zoo' example ClickDOS would put in the temporary file:

C:Zoo -add RAM:temp.zoo RAM:myprog1

C:Zoo -add RAM:temp.zoo RAM:myprog2

C:Zoo -add RAM:temp.zoo RAM:myprog3

3. ClickDOS II then executes the Spawn program and passes to the Spawn program the name of the temporary file and the IO mode it should use to execute the commands. At this point ClickDOS II has finished its part of the job and you can continue to do other things.

4. Spawn opens the temporary file and reads each line executing the command. When it is finished executing all the commands in the file it deletes the temporary file and removes itself from memory.

The List CRA's in ClickDOS II were implemented in this way for two reasons:

1. AmigaDOS has a limit of 256 characters on a command line. Therefore, ClickDOS II could not simply append all of the selected file names and then execute the command as one big long command line. This would be fine if only two or three files are selected, but would crash your machine if a hundred files were selected.

2. This method allows you to have programs which normally do not allow multiple filenames on the command line to act as if they do.

There is, of course, a trade-off for using this method. The command must be executed over and over for each selected file. This means it must be loaded from disk once for every selected file. If you have a hard drive this is no problem. However, if you have only floppies it can be tedious. Floppy disk cache programs, such as FaccII, copying commands to RAM:, and using resident commands can all greatly enhance performance.

There is one final CRA which we have yet to talk about, that is the Pause CRA (%p). The Pause CRA is not really a replacement argument. It is, instead, a way to give yourself messages or warnings when you select a user-defined gadget.

When ClickDOS II finds a Pause CRA in a command's argument list it opens a requester containing the prompt from the Pause CRA. This requester has two gadgets: CANCEL and OK. If you select cancel, execution of the user defined gadget is canceled.

If you select OK, then execution of the command continues. Nothing is substituted for the Pause CRA as it simply removed from the argument list. You could, for example, use a Pause CRA in a user defined gadget which does a very drastic action to give yourself one final way of cancelling the command. Here is a rather extreme example of a Pause CRA:

USER6=DIE;

"CON:0/0/640/200/I'm Killing your hard drive";

dh0:system/format;

DRIVE DH0: NAME Deaddrive NOICONS %p[Format DH0: Are you sure?];

V2.00 ClickDOS II IS NO LONGER DETACHED:

Because of a conflict with the ARP library and the Manx Aztec C detach code, ClickDOS II no longer automatically detaches itself from a CLI when it starts. You must type:

ARUN ClickDOS II NOIO

After careful consideration I found that ARP added enough to the power of ClickDOS II to justify losing the detach feature. For all of you out there who really liked that feature.

-- SORRY! For those who don't know what I'm talking about, don't worry about it.

ARUN is an ARP command. You should use it instead of the AmigaDOS RUN command. If you use the AmigaDOS RUN command you will not be able to close the CLI window you start ClickDOS II from even if you use the >NIL: <NIL: options of RUN. The ARUN command, along with the full release of ARP 1.3 is included with ClickDOS II.

STOPPING ClickDOS II:

OK FINE, BUT HOW DO I GET OUT OF THIS THING: Click the close gadget!

KNOWN BUGS

None are known at this time! However, you should use the RUN action with care.

Appendix A - The Spool Program

Spool is a printer spooling program which should have been included on the ClickDOS II distribution disk. ClickDOS II will automatically start Spool, if it is not already in memory, and turn printing over to it when you use the PRINT action. You can use the SPOOL option in the ClickDOS-Startup file to tell ClickDOS II where to find the Spool program.

NOTE: Spool and ClickDOS II communicate intimately using Inter-Task Messages. You CANNOT substitute a different spooler program for Spool!

Spool records the name and full path of the files sent to it and then reads the files directly from disk and sends them to the printer using the DOS PRT: device. By default the Spool program does very little buffering so it uses only a small amount of memory; Spool is about 8K itself and it opens a 5K buffer. The Spool window is very small and will open in the title bar of the ClickDOS II window. Spool will continuously tell you the name of the file it is currently printing. You can cancel ALL printing by clicking the close gadget of the Spool window.

ClickDOS II will never put more than one copy of Spool in memory. If you use the PRINT action again before Spool has finished printing, ClickDOS II will tell Spool to add the new files names to its queue of files waiting to be printed rather than starting Spool a second time.

Once Spool is started, it will stay in memory even after it has finished printing. When Spool is not printing but sitting idle waiting for something to print, it will display "Waiting..." in its window. Spool is also very nice to use from a CLI. You can spool one or more files to the printer by typing:

SPOOL filename filename filename ...

Spool will start itself and begin printing the file(s). You cannot use wildcards with Spool. If you need to print a large number of files, use ClickDOS II. The nice thing about Spool, which makes it better than something like:

RUN TYPE filename >PRT:

is that if you decide you want to print more files, you do not have to wait for the first file(s) to finish printing. You can simply execute the Spool command again with more file names.

Every time Spool is started, it looks to see if there is a copy of itself already in memory. If it finds a copy of itself, it simply tells the other copy to add the new files to its queue of files waiting to be printed. If it does not find itself in memory, then it prints the files itself.

As stated above, Spool normally does very little buffering. This means that the disk which Spool is reading the file from, must stay in the drive. This is not a problem for hard drives but can be annoying for floppies. You can have ClickDOS II increase the size of the buffer. Spool uses with the SBUFF option in the ClickDOS-Startup files discussed below. If you make the buffer big enough, Spool will be able to read the entire file into memory so that the disk will not have to stay in the drive. You can change the size of the buffer Spool uses "on the fly", even when something is printing, by using the following command at a CLI:

SPOOL -bXXXX

where XXXX is the size of the buffer you want Spool to use, in bytes. You can issue this command to change the buffer size, even if the printing was started from ClickDOS II. The print buffer cannot be less than 1K (1024) and must be a multiple of 1024 (i.e. 1024 must divide into the value evenly.)

There will be times when Spool will have problems: not enough memory, printer not ready, etc. Unfortunately Spool's window is too small to print an error message. Therefore, when Spool has problems it will flash the screen and display in its window the message:

FATAL ERROR # XX

where XX is one of the following numbers:

1 - A buffer cannot be allocated for spooling. This usually means you are low on memory. Spool will shut down gracefully after this error. Try starting Spool with a smaller buffer. You can do this AND spool a file at the same time like this:

SPOOL -bXXXX filename

2 - Spool cannot write to the printer for some reason. Spool will shut down gracefully after this error. Check your printer. Make sure you are not trying to print with two different programs at the same time.

3 - A file you wanted to print could not be open for some reason. The file may have an error in it. If you spooled the file from a CLI try again, but specifying the full path to the file, for example: **SPOOL dh0:myfiles/myfile**

This is not a fatal error, even though Spool will report it as such. Spool will simply go to the next file in its queue and continue printing.

4 - Can't allocate a required internal structure. If you get this error you probably are running low on memory. Spool will shut down gracefully.

5 - Cannot re-size the buffer. Spool received a request to re- size its print buffer, but after closing the old one, it cannot open a new buffer of the requested size. Spool will shut down gracefully. Spool closes the old buffer before opening the new buffer, so that both buffers do not have to be allocated in memory at the same time. Try restarting Spool with a smaller buffer.

V2.00 Appendix B - ClickDOS II temporary files

ClickDOS II will create temporary files on RAM: if you use the RUN option or if you use the print-window-contents option of TYPE. These temporary files have names of the form:

cd_XXXXXXX.tmp

where XXXXXXXX is a number. These files should always be deleted by the programs SPOOL or SPAWN. These files may be left in RAM:, however, if an error occurs and ClickDOS II cannot start the SPOOL or SPAWN programs.

